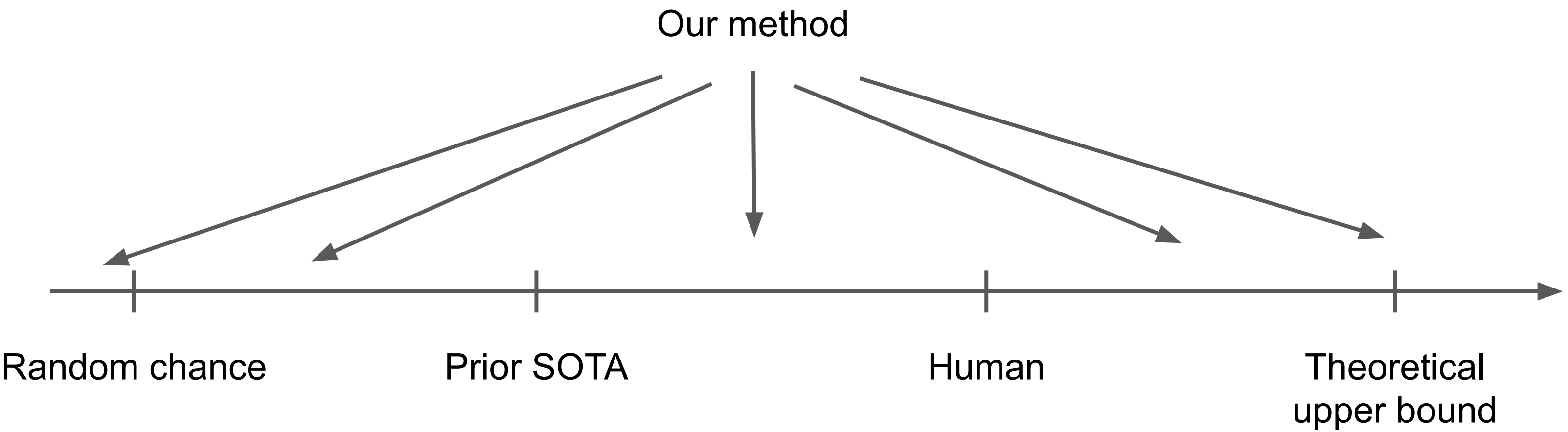
Speaker: Seong Joon Oh (NAVER)

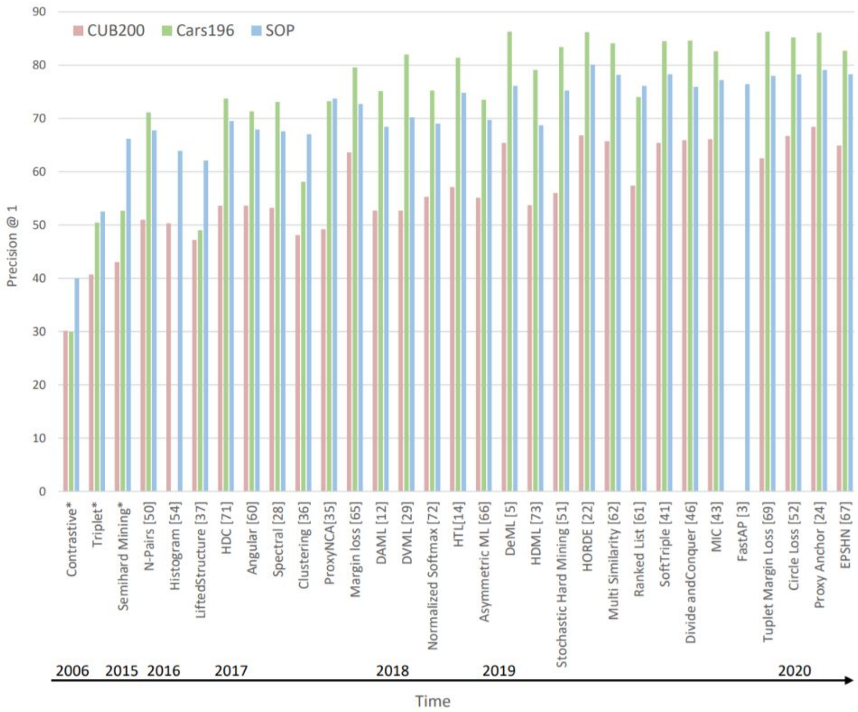
Evaluating weakly-supervised models

Why do we do evaluation?

It enables ranking.



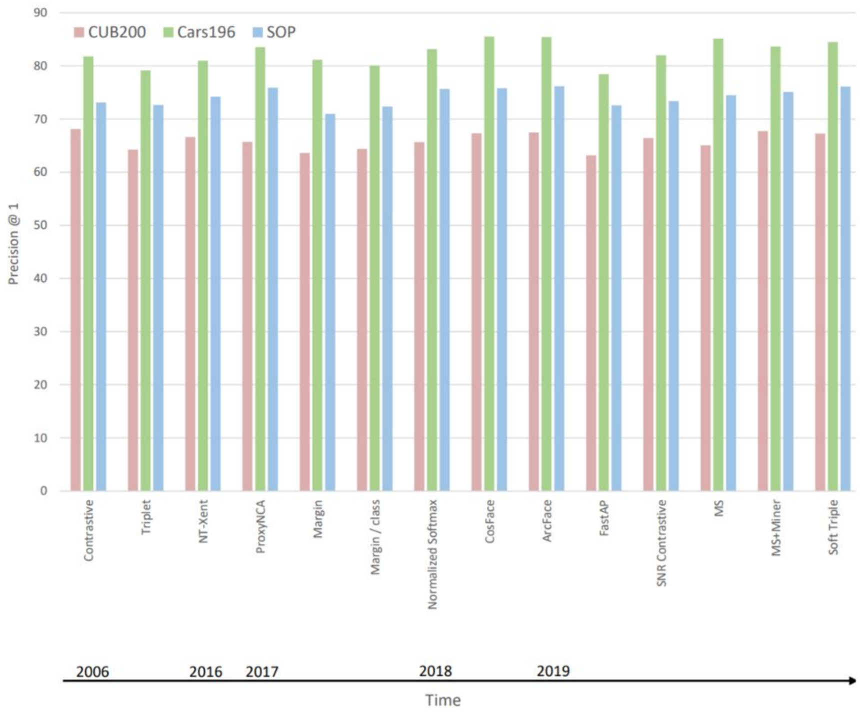
What are the costs of wrong evaluation?



(a) The trend according to papers

Musgrave et al. A Metric Learning Reality Check. ECCV'20.

What are the costs of wrong evaluation?



(b) The trend according to reality

Musgrave et al. A Metric Learning Reality Check. ECCV'20.

What are the costs of wrong evaluation?

Researchers

- 4+ years efforts put into pursuing the wrong metric.
- Opportunity cost: what if they have worked on other “real” challenges?

Practitioners

- Misinformed selection of methods based on the wrong ranking.
- Cost of neglecting a simple solution that works equally well.

Musgrave et al. A Metric Learning Reality Check. ECCV’20.

Similar “evaluation scandals”
in many CV & ML tasks.

Face detection: Mathias et al. Face Detection without Bells and Whistles. ECCV’14.

Zero-shot learning: Xian et al. Zero-Shot Learning-The Good, the Bad and the Ugly. CVPR’17.

Semi-supervised learning: Oliver et al. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. NeurIPS’18.

Unsupervised disentanglement: Locatello et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. ICML’19.

Image classification: Recht et al. Do ImageNet Classifiers Generalize to ImageNet? ICML’19.

Scene text recognition: Baek et al. What Is Wrong with Scene Text Recognition Model Comparisons? Dataset and Model Analysis. ICCV’19.

Weakly-supervised object localization: Choe et al. Evaluating Weakly-Supervised Object Localization Methods Right. CVPR’20.

Deep metric learning: A Metric Learning Reality Check. ECCV’20.

Natural language QA: Lewis et al. Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets. ArXiv’20.

Recipes for wrong evaluation.

1. Everyone writes their own evaluation metric code.

There are non-trivial code-level details in some evaluation metrics.

E.g. For computing average precision (AP), how do you handle precision values for high-confidence bins where

$$\text{precision} := \frac{\text{true positive}}{\#\text{positive prediction}} = \frac{0}{0}$$

http://host.robots.ox.ac.uk/pascal/VOC/
Choe et al. Evaluating Weakly-Supervised Object Localization Methods Right. CVPR’20.

2. Confound multiple factors when comparing methods.

k	1	10	100	1000	NMI
Histogram [34]	63.9	81.7	92.2	97.7	-
Binomial Deviance [34]	65.5	82.3	92.3	97.6	-
Triplet Semi-hard [25, 29]	66.7	82.4	91.9	-	<u>89.5</u>
LiftedStruct [22, 29]	62.5	80.8	91.9	-	88.7
StructClustering [29]	67.0	83.7	<u>93.2</u>	-	<u>89.5</u>
N-pairs [28]	67.7	83.8	93.0	<u>97.8</u>	88.1
HDC [41]	<u>69.5</u>	<u>84.4</u>	92.8	97.7	-
Margin	**72.7**	**86.2**	**93.8**	**98.0**	**90.7**

Improvements come from the loss function?

Musgrave et al. A Metric Learning Reality Check. ECCV'20.
Wu et al. Sampling Matters in Deep Embedding Learning. ICCV'17.

2. Confound multiple factors when comparing methods.

Architecture	k	1	10	100	1000	NMI
GoogleNet	Histogram [34]	63.9	81.7	92.2	97.7	-
GoogleNet	Binomial Deviance [34]	65.5	82.3	92.3	97.6	-
Inception-BN	Triplet Semi-hard [25, 29]	66.7	82.4	91.9	-	89.5
Inception-BN	LiftedStruct [22, 29]	62.5	80.8	91.9	-	88.7
Inception-BN	StructClustering [29]	67.0	83.7	93.2	-	89.5
Inception-BN	N-pairs [28]	67.7	83.8	93.0	97.8	88.1
GoogleNet	HDC [41]	69.5	84.4	92.8	97.7	-
ResNet50	**Margin**	**72.7**	**86.2**	**93.8**	**98.0**	**90.7**

Or from the architecture?

Musgrave et al. A Metric Learning Reality Check. ECCV'20.
Wu et al. Sampling Matters in Deep Embedding Learning. ICCV'17.

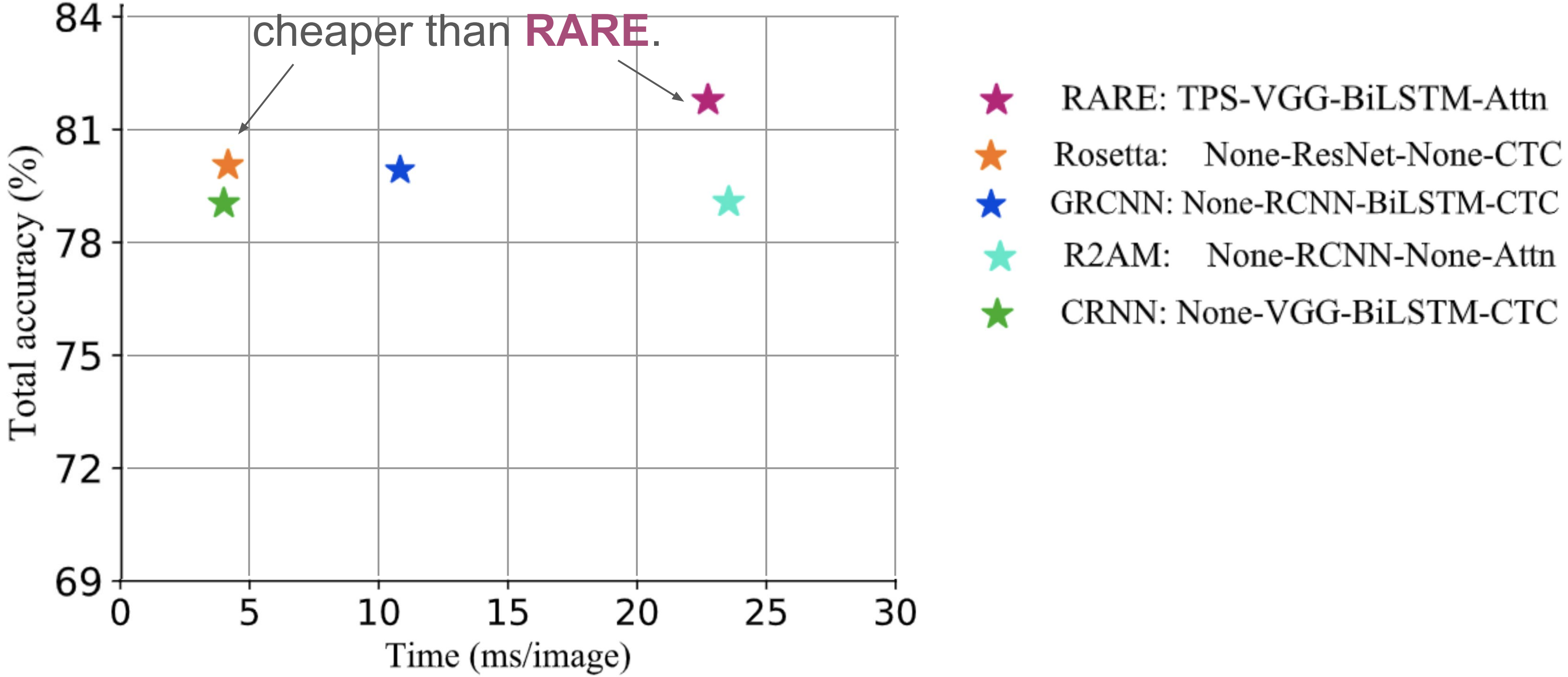
3. Hide extra resources needed to make improvements.

RARE looks like an obvious choice over **Rosetta**.

Baek et al. What Is Wrong with Scene Text Recognition Model Comparisons? Dataset and Model Analysis. ICCV'19.

3. Hide extra resources needed to make improvements.

But **Rosetta** is much cheaper than **RARE**.



Baek et al. What Is Wrong with Scene Text Recognition Model Comparisons? Dataset and Model Analysis. ICCV’19.

4. Train and test samples overlap.

Dataset	% Answer overlap	% Question overlap
Natural Questions	63.6	32.5
TriviaQA	71.7	33.6
WebQuestions	57.9	27.5

Fraction of test sets overlapping with the training set for natural language Q & A task.

Lewis et al. Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets. ArXiv'20.

4. Train and test samples overlap.

Model		Open Natural Questions			
		Total	Question Overlap	Answer Overlap Only	No Overlap
Closed book	T5-11B+SSM	36.6	77.2	22.2	9.4
	BART	26.5	67.6	10.2	0.8
Nearest Neighbor	Dense	26.7	69.4	7.0	0.0
	TF-IDF	22.2	56.8	4.1	0.0

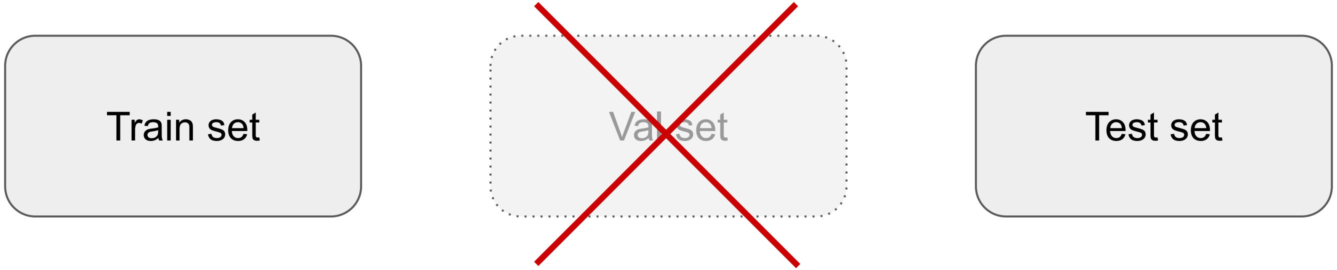
Model performances in different partitions of the test set.

Models have solved the task by **memorising**, rather than by **generalising**.

Lewis et al. Question and Answer Test-Train Overlap in Open-Domain Question Answering Datasets. ArXiv'20.

5. Lack of validation set

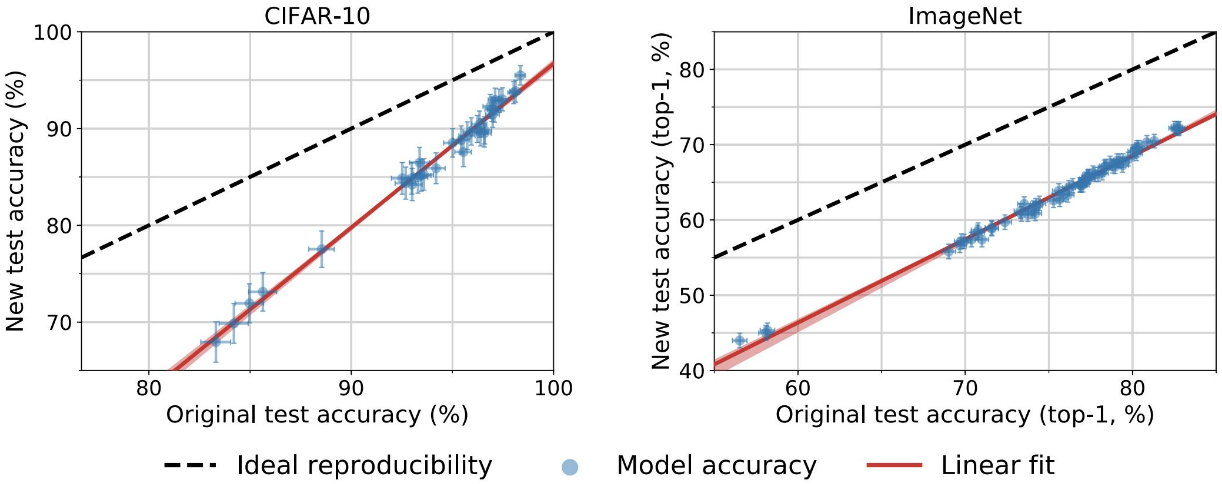
CIFAR and ImageNet classification benchmarks lack validation sets.



Models have made their design choices & HP tuning over the test set.

Recht et al. Do ImageNet Classifiers Generalize to ImageNet? ICML’19.

5. Lack of validation set



Models show dropped performances on new samples from the same distribution.

- Evidence of “overfitting” the design choices to the test set over time.

Recht et al. Do ImageNet Classifiers Generalize to ImageNet? ICML’19.

This talk:
What can go wrong with evaluation?

This talk:

~~What can go wrong with evaluation?~~

What can go wrong with **weakly-supervised X** evaluation?

Added complications for WSX evaluation.

Train/val/test splits for regular ML task.

Train set	Val set	Test set
Model fitting.	Model design choices. Tuning HPs.	Report final numbers. Comparison across methods.

Added complications for WSX evaluation.

Train/val/test splits for weakly-supervised X task.

Train set (Weak sup)	Val set (Full sup)	Test set (Full sup)
Model fitting, using weak supervision.	??? (no agreement on how to use it)	Report final numbers. Comparison across methods.

Added complications for WSX evaluation.

Train/val/test splits for weakly-supervised X task.

Train set
(Weak sup)

Val set
(Full sup)

Test set
(Full sup)

“Weakly-supervised” method is supposed to use this set **ONLY**.

Added complications for WSX evaluation.

Train/val/test splits for weakly-supervised X task.

Train set
(Weak sup)

Val set
(Full sup)

Test set
(Full sup)

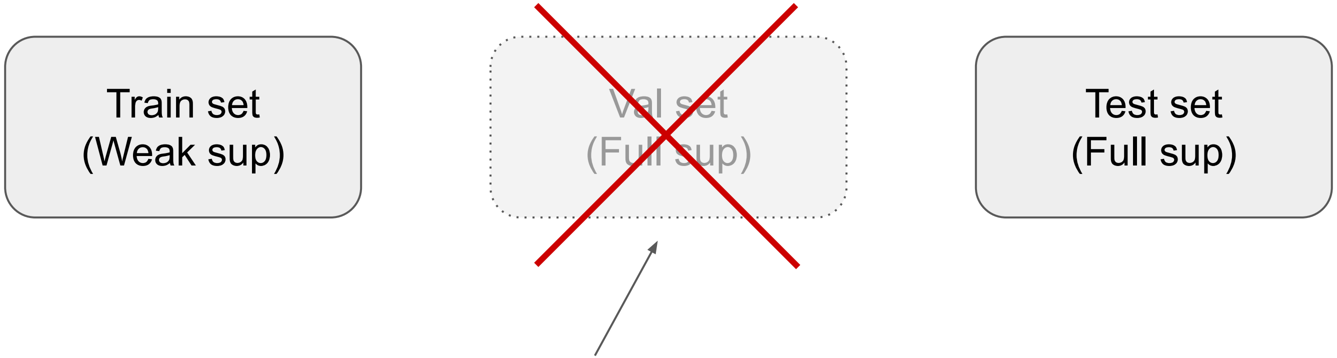
Usually used for tuning HPs.

Lack of unified agreement on “how to use”.

Some methods extensively make use of
val set for HP search (e.g. grid search)
→ Unfair !

Added complications for WSX evaluation.

Train/val/test splits for weakly-supervised X task.



Even worse, there is no val set
in many WSX benchmarks.

Added complications for WSX evaluation.

Train/val/test splits for weakly-supervised X task.

Train set
(Weak sup)

~~Val set~~
~~(Full sup)~~

Test set
(Full sup)

And people tune their HPs
over the test set !

Added complications for WSX evaluation.

Correct evaluation is even more tricky for WSX.

1. Implicit tuning on the test set (problem shared by regular ML tasks).

2. **Implicit use of full supervision (specific to WSX tasks).**

Added complications for WSX evaluation.

These evaluation issues with WSX are not widely known yet.

Train set
(Weak sup)

~~Val set~~
~~(Full sup)~~

Test set
(Full sup)

Many researchers and practitioners are still misinformed by wrong evaluation results.

This is the first time the issue is discussed in a tutorial.

Case study: Weakly-supervised object localization.

WSOL is the "minimal working example" for the WSX evaluation issues.

Same problem in WSSS (semantic segmentation), WSOD (object detection), WSIS (instance segmentation), SSL (semi-supervised learning), UD (unsupervised disentanglement), ZSL (zero-shot learning), etc.

Other motivations

- Popularity: 100+ papers in the last 5+ years.
- Applicability: Ingredient for other WSX tasks.

CVPR 2020

Evaluating Weakly-Supervised Object Localization Right.


Junsuk Choe*
NAVER


Seong Joon Oh*
NAVER


Seungho Lee
Yonsei Univ.


Sanghyuk Chun
NAVER


Zeynep Akata
University of Tübingen


Hyunjung Shim
Yonsei Univ.

* Equal contribution

What is WSOL?

WSOL = Weak supervision + Object localization.

- What is object localization?
- What type of weak supervision?

What is object localization?

What's in the image?

Single-label classification

- One class per image.

What is object localization?

Where’s the cat?

Object localization

- One class per image.
- Class is known (there’s a cat).
- *Point me out where the cat is.*

Output format:

- Point
- Box
- **Mask** (default mode in this talk).

Object localization ≠ Semantic segmentation

Classify each pixel in image:

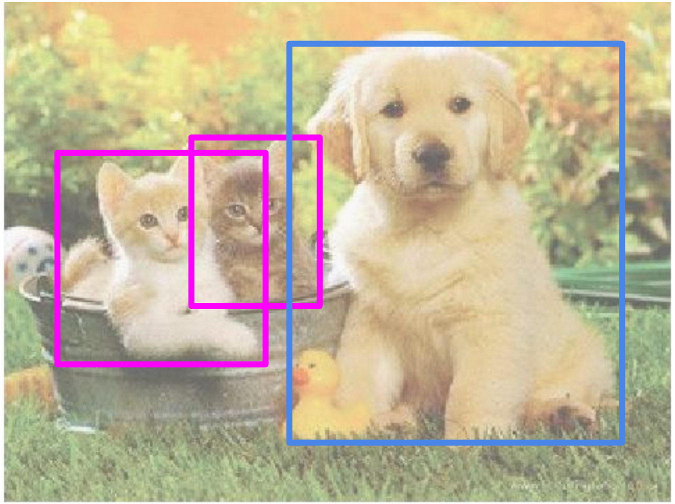
Semantic segmentation

Semantic segmentation setup:

- Multiple object classes per image.
- GT class not given.
- Pixel-wise labelling.

Object localization ≠ Object detection

Box all instances & classify them:

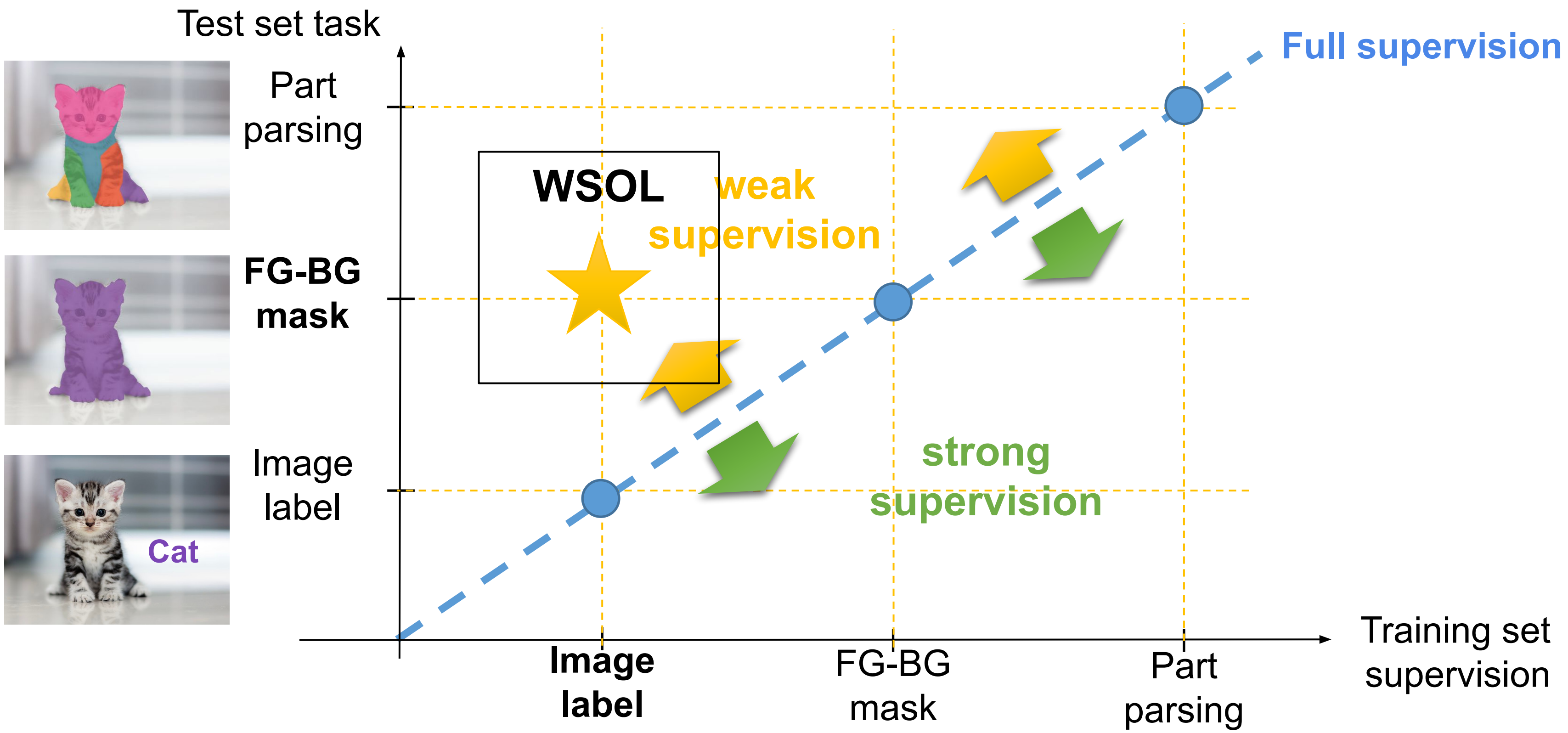


Object detection

Object detection setup:

- Multiple object classes per image.
- Need to separate instances.
- GT class not given.

What do you mean by weak supervision?



WSOL methods:
How are they trained & evaluated?

Class Activation Mapping (CAM) example.

Train set
(Weak sup)

~~Val set~~
~~(Full sup)~~

Test set
(Full sup)

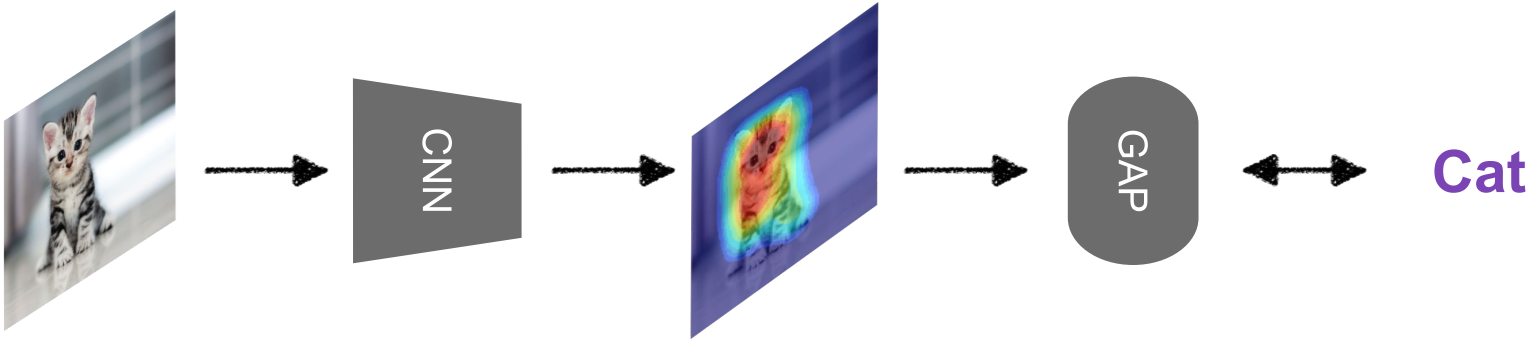
Zhou et al. Learning deep features for discriminative localization. CVPR'16.

CAM during training (weak annotation)

Train set
(Weak sup)

~~Val set
(Full sup)~~

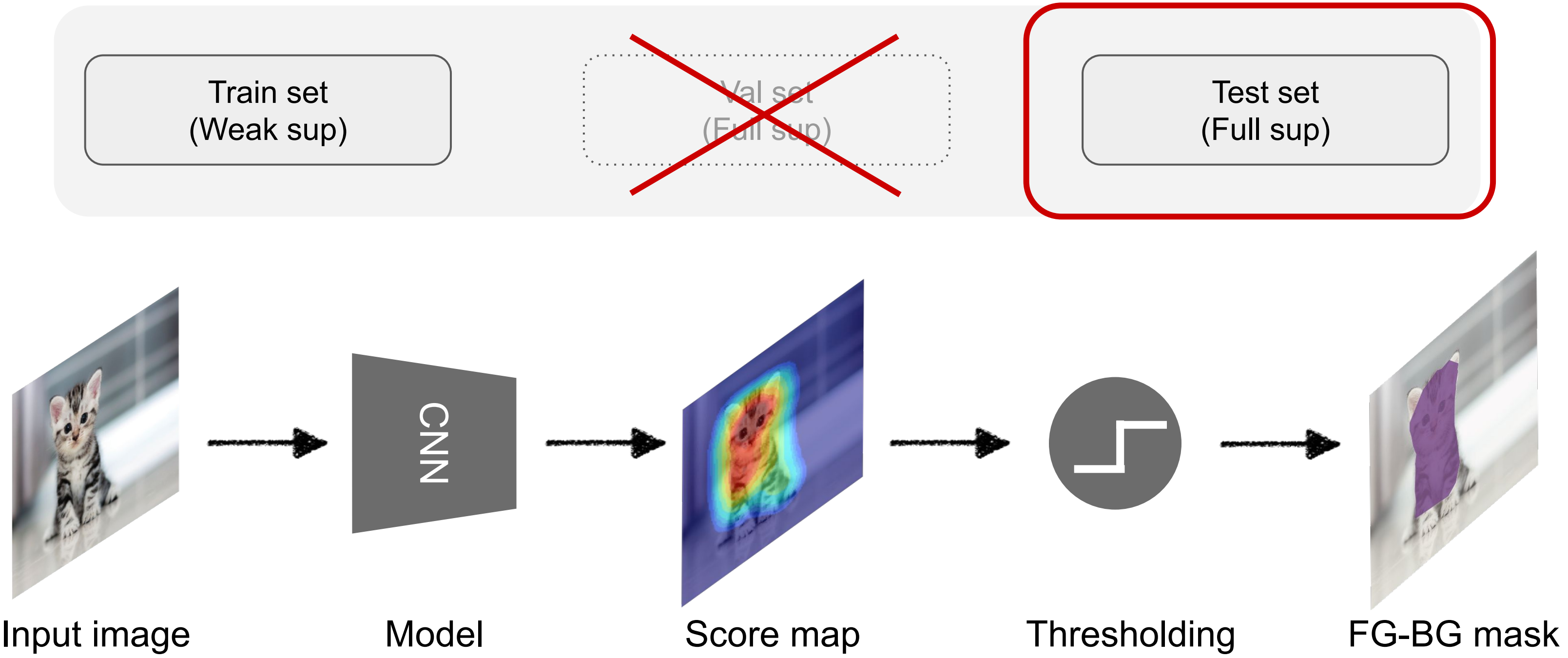
Test set
(Full sup)



Input image | Model | Score map | Spatial pooling | Class label

Zhou et al. Learning deep features for discriminative localization. CVPR’16.

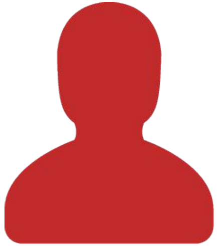
CAM during evaluation (full annotation)



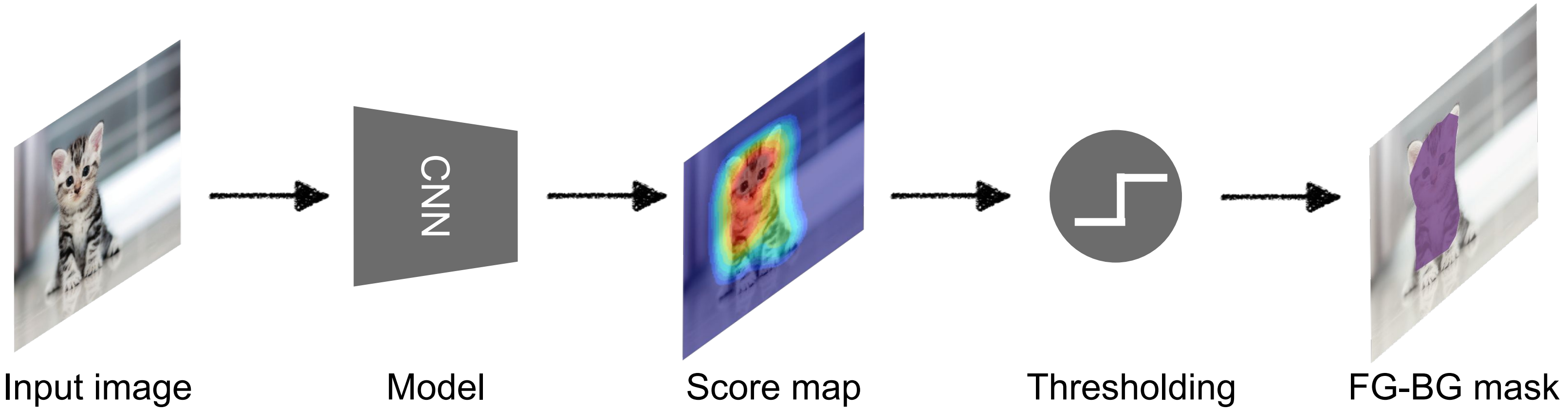
Zhou et al. Learning deep features for discriminative localization. CVPR’16.

CAM does not use any full supervision, does it?

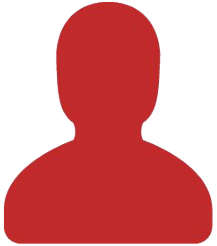
How things can go wrong (**Version 1**)



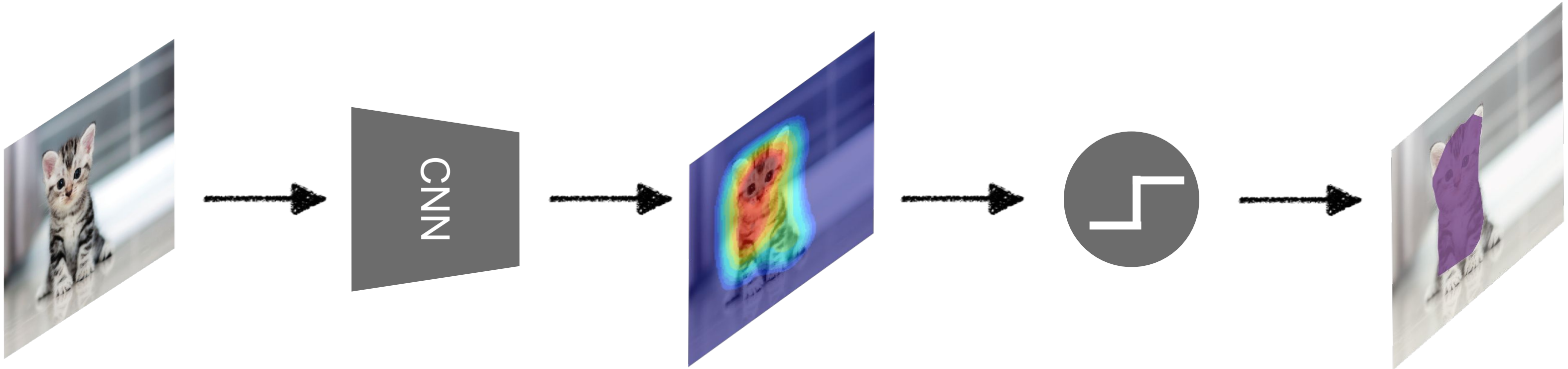
Which threshold do we choose?



How things can go wrong (**Version 1**)

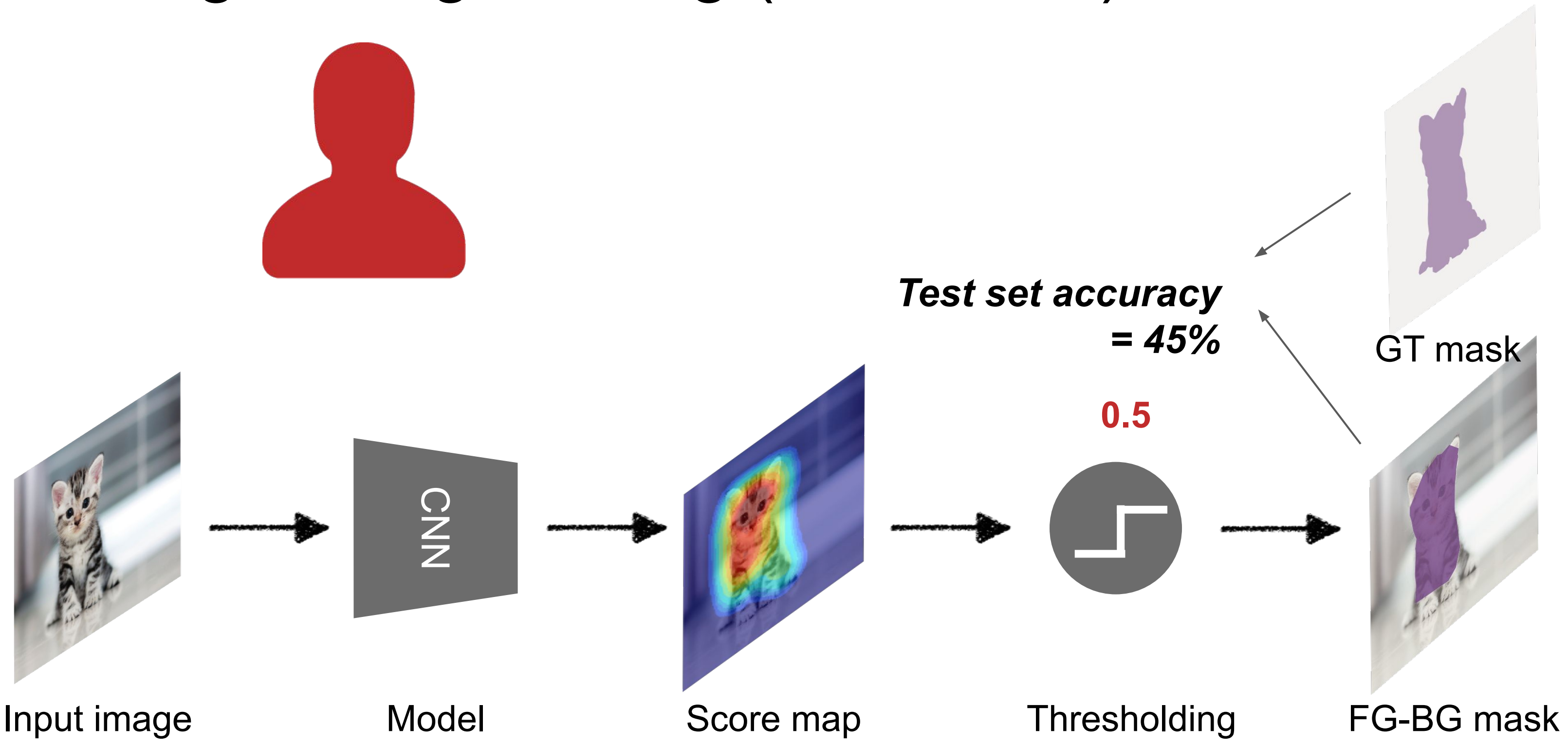


Let's see… I'll choose 0.5 because why not?

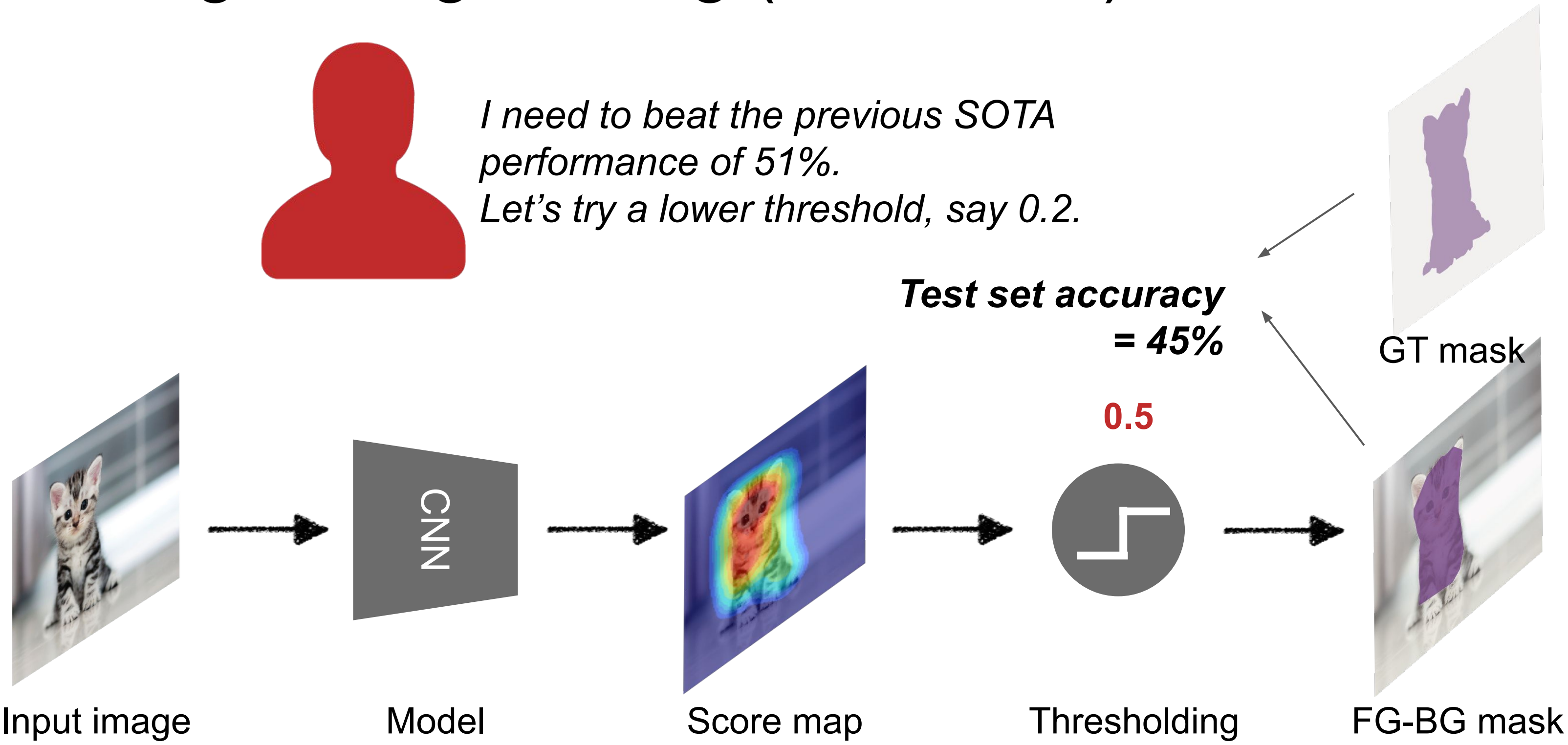


Input image Model Score map Thresholding FG-BG mask

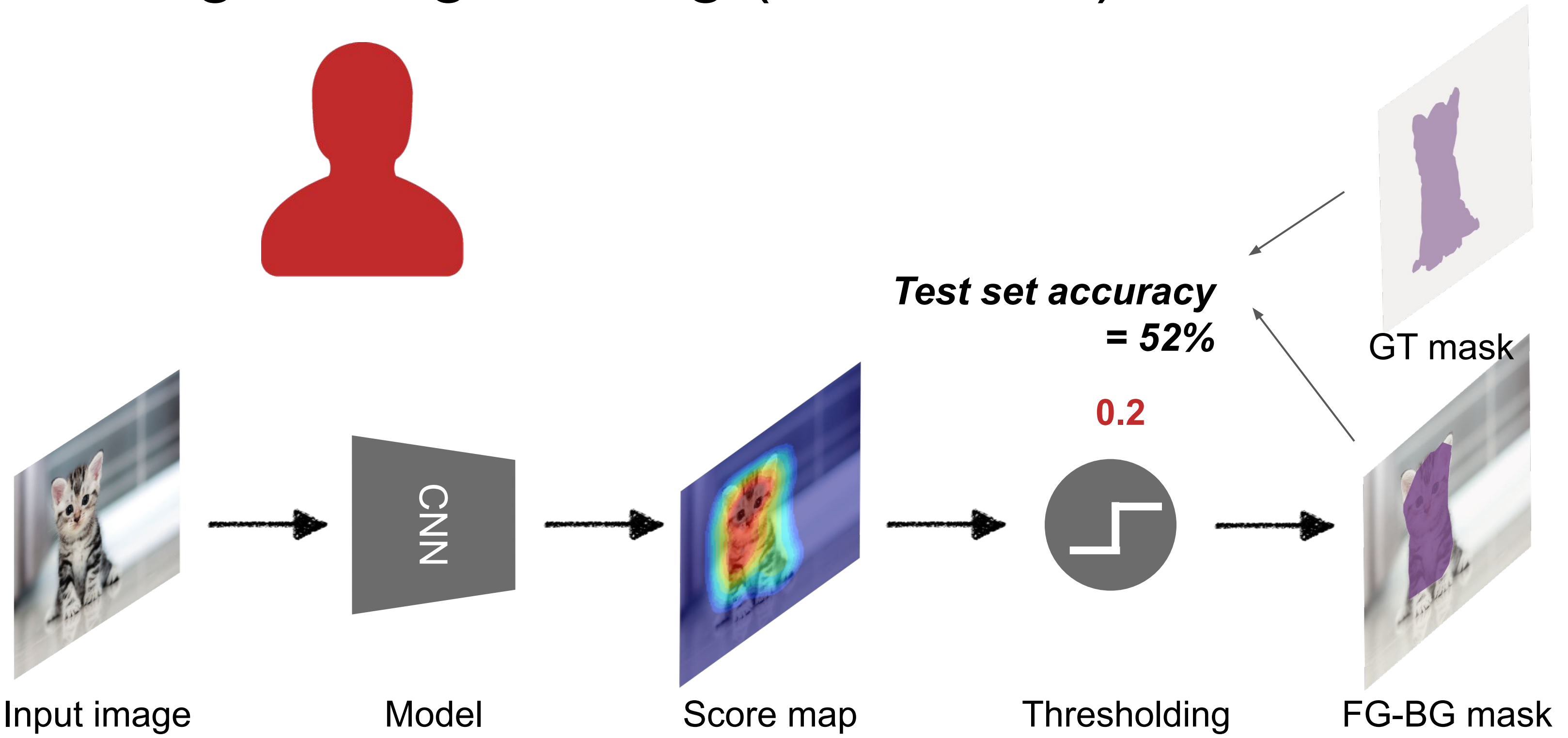
How things can go wrong (Version 1)



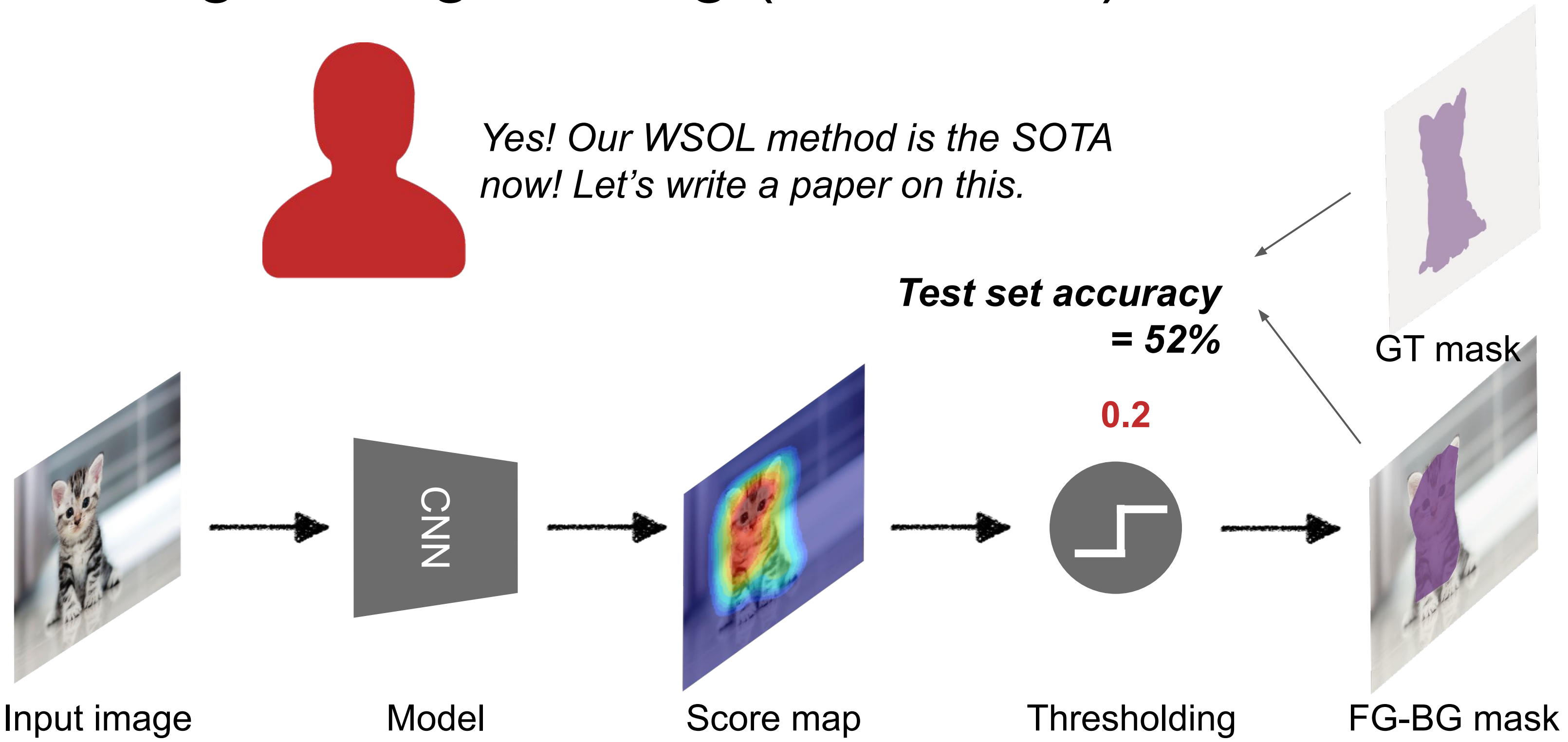
How things can go wrong (Version 1)



How things can go wrong (**Version 1**)

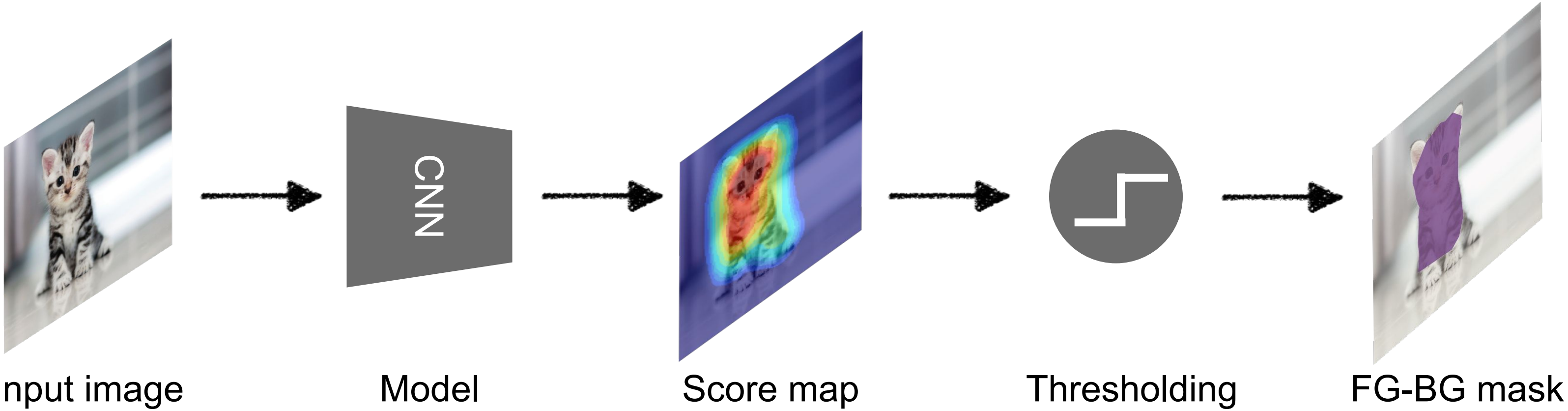


How things can go wrong (Version 1)



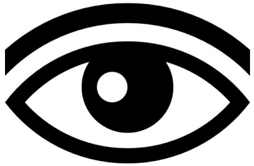
How things can go wrong (Version 2)

Let's not tune the HPs on the test set.
Let's never touch the full supervision.

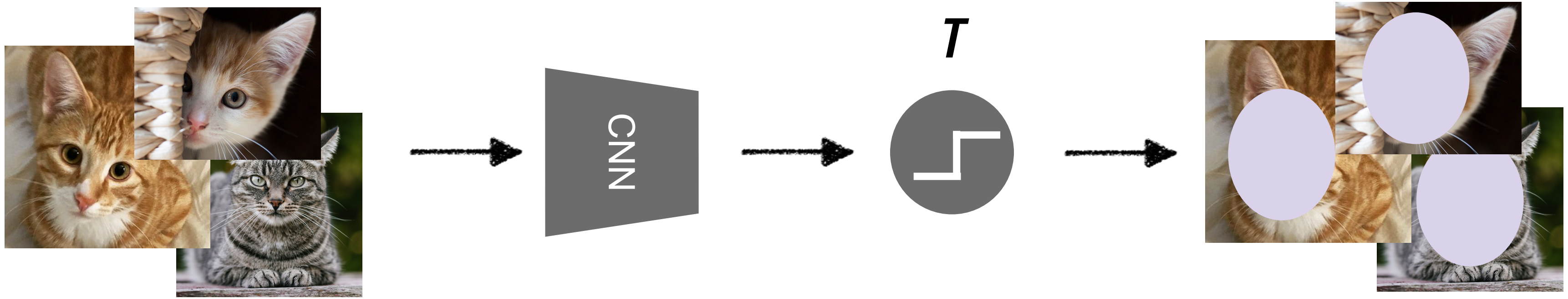


How things can go wrong (**Version 2**)

Instead, let's inspect a few outputs in the training set.



Human inspection



A few training samples

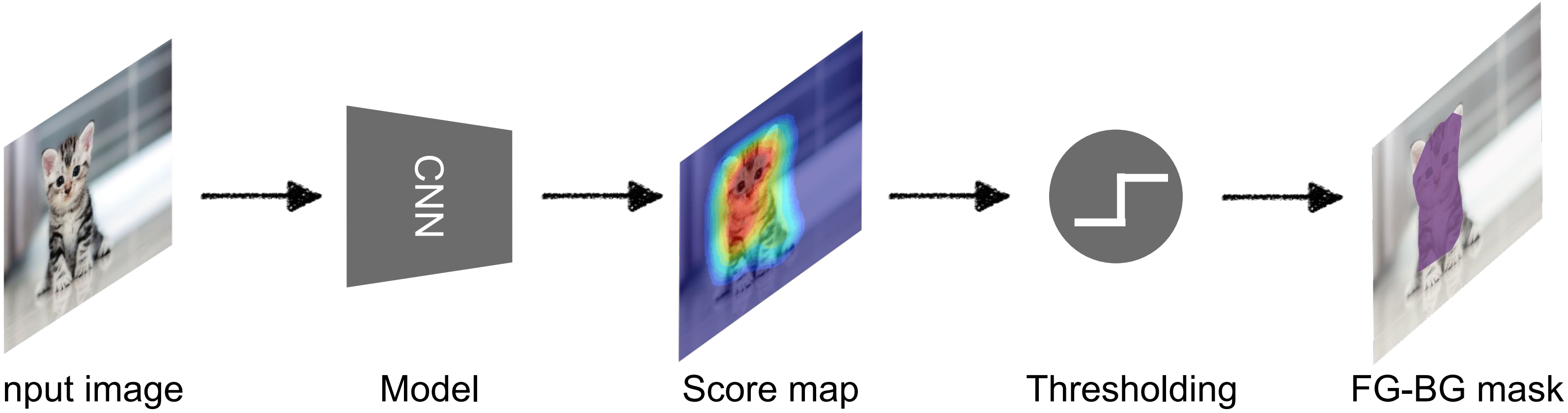
Model

Thresholding

FG-BG masks

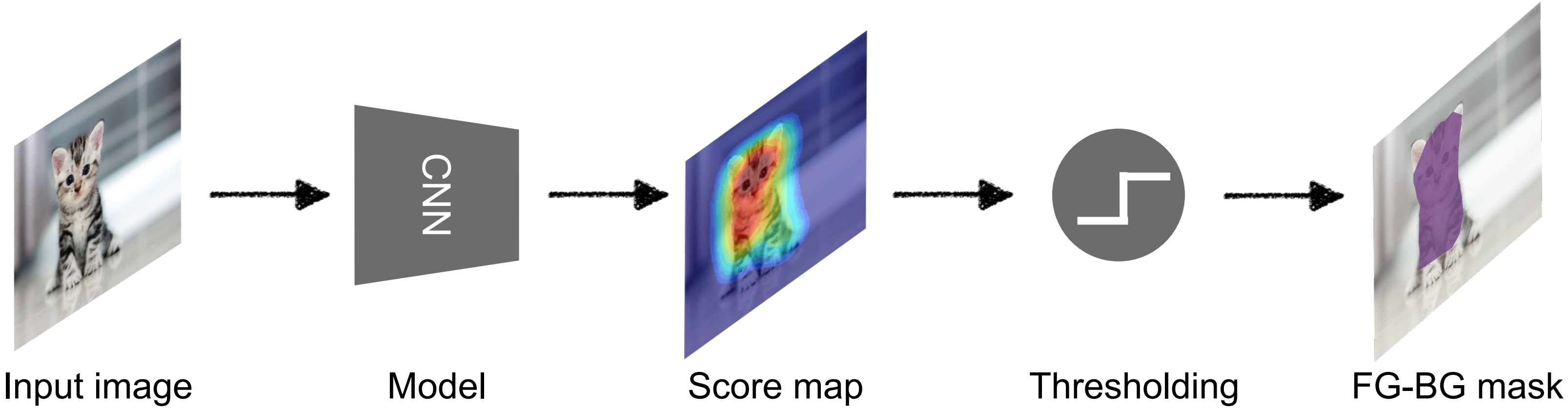
How things can go wrong (**Version 3**)

Human-in-the-loop is also violating the weak-supervision policy.

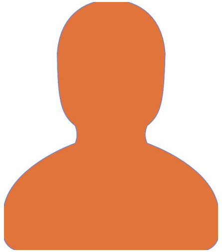


How things can go wrong (**Version 3**)

We are going to adopt whatever HPs previous papers have been using.

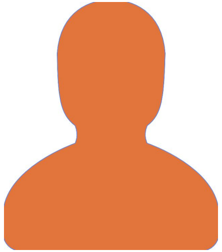


How things can go wrong (**Version 4**)



(Black magic happening)

How things can go wrong (**Version 4**)



In paper:
“We use threshold 0.5 [full stop]”

Common strategies for a good WSOL performance.

(1) Introduce many HPs.

(2) Tune the HPs with one of the following strategies:

- **Version 1: Validation on test set**
- **Version 2: Human-in-the-loop**
- **Version 3: “It’s not our fault”**
- **Version 4: Black magic**

Arguably, versions 1-4 are different versions of implicit full supervision.

How methods tune their HPs.

WSOL method	Hyperparameters	How to tune them
CAM, CVPR'16	Threshold / Learning rate / Feature map size	**(4) Black magic**
HaS, ICCV'17	Threshold / Learning rate / Feature map size / Drop rate / Drop area	**(2) Human in the loop, (3) “Not our fault”**
ACoL, CVPR'18	Threshold / Learning rate / Feature map size / Erasing threshold	**(1) Tune HP with full sup**
SPG, ECCV'18	Threshold / Learning rate / Feature map size / Threshold 1L / Threshold 1U / Threshold 2L / Threshold 2U / Threshold 3L / Threshold 3U	**(1) Tune HP with full sup**
ADL, CVPR'19	Threshold / Learning rate / Feature map size / Drop rate / Erasing threshold	**(1) Tune HP with full sup**
CutMix, ICCV'19	Threshold / Learning rate / Feature map size / Size prior / Mix rate	**(3) “Not our fault”**

Implicit full supervision in other WSX tasks.

- **WSSS**: “those pixels belonging to top 20% of the largest value (a fraction suggested by [14, 33]) in the heatmap are considered as foreground object regions” (Huang et al. CVPR’18) **“It’s not our fault”**
- **WSOD**: “The constant τ_0 in Eq. (17) and the threshold T_c [are] empirically set to 0.5 and 0.1, respectively” (Li et al. ICCV’19) **Human-in-the-loop?**
- **WSOD**: “We empirically set the hyperparameter β to 0.8.” (Wang et al. ICJAI’18). **Human-in-the-loop?**
- **WSIS**: “[...] γ [...] is set to 10 when training, and reduced to 5 at inference [...] t [...] is fixed to 256 [...] β [...] is set to 10 [...] D is 100 [...].” (Ahn et al. CVPR’19) **Black magic**

And many others.

We are not trying to
blame the researchers.

We argue instead that
extra information is inevitable for WSX.

WSOL is ill-posed Choe et al. CVPR'20

Pathological case:

A class (e.g. duck) correlates better with a BG concept (e.g. water) than a FG concept (e.g. feet).

Then, WSOL is not solvable even with infinite supply of training data.

The four strategies then make sense!

- **Version 1: Validation on test set**
- **Version 2: Human-in-the-loop**
- **Version 3: “It’s not our fault”**
- **Version 4: Black magic**

For fair comparison, we need to let methods use

- Equal amount of extra information
- Identical HP search strategy with same amount of computational budget

Solution: Introduce the validation set!

Roadmap for the rest of the talk.

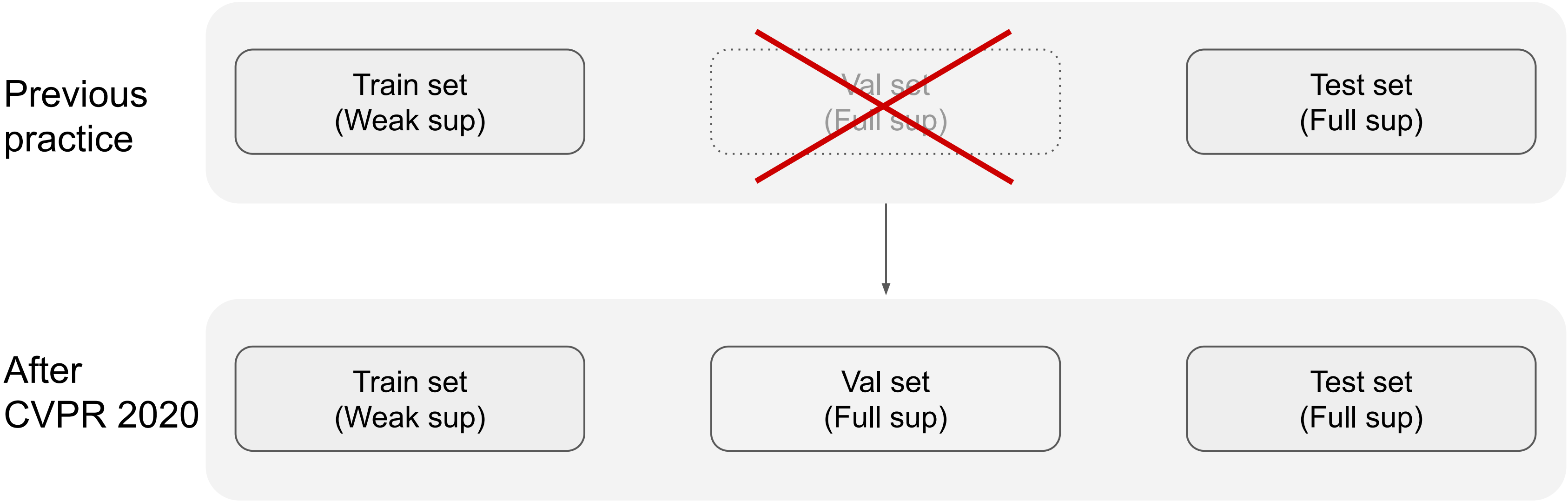
1. Introduce validation set in WSOL (legalise the use of extra information).

2. Re-rank recent WSOL methods under the new evaluation protocol.

3. Future directions for WSOL and WSX, given the inevitability of extra info.

Introducing the validation set for WSOL.

Introducing the “validation set” for WSOL.



We let WSOL methods search HPs over the identical val set.

- Ensures equal amount of extra information for each method.

Existing WSOL benchmarks and datasets.

Dataset	Training set (Weak sup)	Validation set (Full sup)	Test set (Full sup)
ImageNet	✓	✗ ImageNetV2; no full sup.	✓
CUB	✓	✗ No images, nothing.	✓

Proposed WSOL benchmarks and datasets.

Dataset	Training set (Weak sup)	Validation set (Full sup)	Test set (Full sup)
ImageNet	✓	✓ Annotate boxes on ImageNetV2.	✓
CUB	✓	✓ Collect images; Annotate boxes.	✓
OpenImages	✓ Curate OpenImages30K	✓ Curate OpenImages30K	✓ Curate OpenImages30K

Choe et al. Evaluating Weakly-Supervised Object Localization Methods Right. CVPR 2020.

Fair algorithm, fair budget, fair resource.

WSOL method	Hyperparameters	How to tune them
CAM, CVPR'16	Threshold / Learning rate / Feature map size	**Version 4**
HaS, ICCV'17	Threshold / Learning rate / Feature map size / Drop rate / Drop area	**Version 2, Version 3**
ACoL, CVPR'18	Threshold / Learning rate / Feature map size / Erasing threshold	**Version 1**
SPG, ECCV'18	Threshold / Learning rate / Feature map size / Threshold 1L / Threshold 1U / Threshold 2L / Threshold 2U / Threshold 3L / Threshold 3U	**Version 1**
ADL, CVPR'19	Threshold / Learning rate / Feature map size / Drop rate / Erasing threshold	**Version 1**
CutMix, ICCV'19	Threshold / Learning rate / Feature map size / Size prior / Mix rate	**Version 3**

Previous search strategies

Fair algorithm, fair budget, fair resource.

WSOL method	Hyperparameters	How to tune them
CAM, CVPR'16	Threshold / Learning rate / Feature map size	**Random search on val set, 30 iterations**
HaS, ICCV'17	Threshold / Learning rate / Feature map size / Drop rate / Drop area	**Random search on val set, 30 iterations**
ACoL, CVPR'18	Threshold / Learning rate / Feature map size / Erasing threshold	**Random search on val set, 30 iterations**
SPG, ECCV'18	Threshold / Learning rate / Feature map size / Threshold 1L / Threshold 1U / Threshold 2L / Threshold 2U / Threshold 3L / Threshold 3U	**Random search on val set, 30 iterations**
ADL, CVPR'19	Threshold / Learning rate / Feature map size / Drop rate / Erasing threshold	**Random search on val set, 30 iterations**
CutMix, ICCV'19	Threshold / Learning rate / Feature map size / Size prior / Mix rate	**Random search on val set, 30 iterations**

CVPR’20: Unified search algorithm

Unifying metrics, datasets, and architectures.

Metrics →	Top1-Loc														GT-known			
Datasets →	ImageNet								CUB						ImageNet			
Architectures →	V	I	R	A	G	N	S	M	V	I	R	G	S	M	V	I	A	G
CAM CVPR'16	42.8	-	46.3	36.3	43.6	34.5	-	41.7	37.1	43.7	49.4	41.0	42.7	43.7	-	62.7	55.0	58.7
HaS ICCV'17	-	-	-	37.7	45.5	-	-	41.9	-	-	-	-	-	44.7	-	-	58.7	60.6
ACoL CVPR'17	45.8	-	-	-	46.7	-	-	-	45.9	-	-	-	-	-	-	-	-	63.0
SPG ECCV'18	-	48.6	-	-	-	-	-	-	-	46.6	-	-	-	-	-	64.7	-	-
ADL CVPR'19	44.9	48.7	-	-	-	-	48.5	43.0	52.4	53.0	-	-	62.3	47.7	-	-	-	-
CutMix ICCV'19	43.5	-	47.3	-	-	-	-	-	-	52.5	54.8	-	-	-	-	-	-	-

Reported results in existing papers

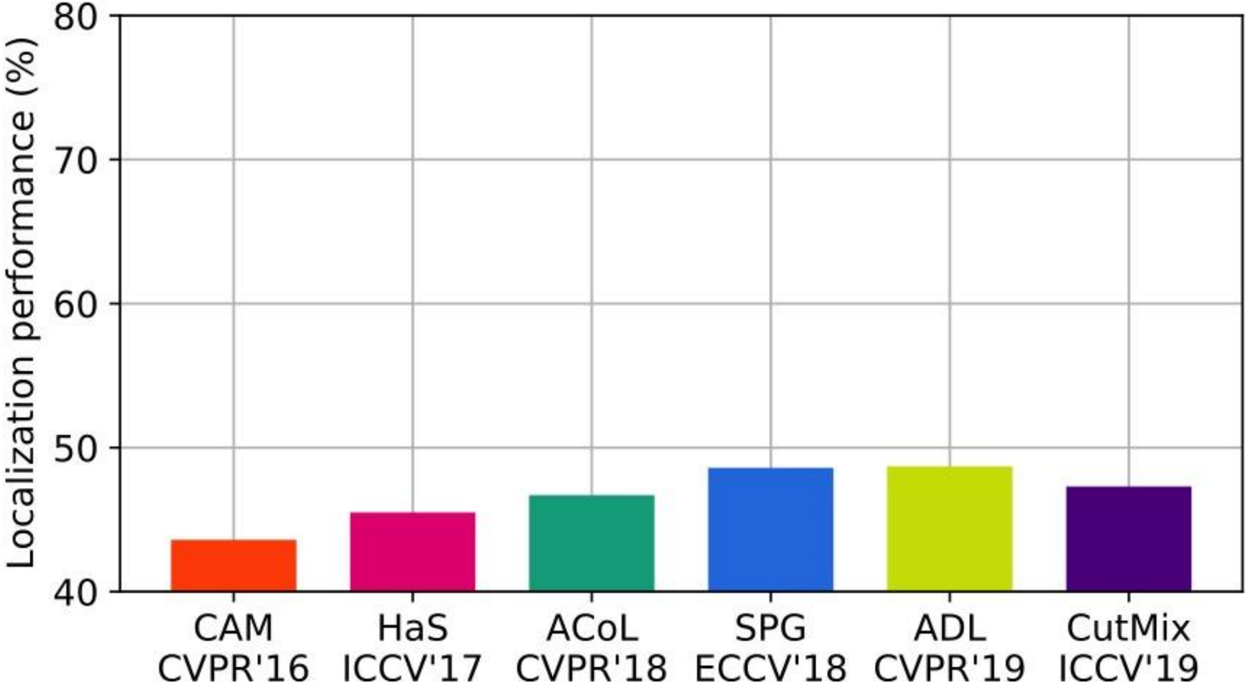
Unifying metrics, datasets, and architectures.

Dataset →	ImageNet (MaxBoxAccV2)				CUB (MaxBoxAccV2)				OpenImages (PxAP)				Total
Architecture →	V	I	R	Mean	V	I	R	Mean	V	I	R	Mean	Mean
CAM CVPR'16	60.0	63.4	63.7	62.4	63.7	56.7	63.0	61.1	58.3	63.2	58.5	60.0	61.2
HaS ICCV'17	60.6	63.7	63.5	62.6	63.7	53.4	64.7	60.6	58.1	58.1	55.9	57.4	60.2
ACoL CVPR'17	57.4	63.7	62.3	61.2	57.4	56.2	66.5	60.0	54.3	57.2	57.3	56.3	59.2
SPG ECCV'18	59.9	63.3	63.3	62.2	56.3	55.9	60.4	57.5	58.3	62.3	56.7	59.1	59.6
ADL CVPR'19	59.8	61.4	63.7	61.7	66.3	58.8	58.4	61.1	58.7	56.8	55.2	56.9	59.9
CutMix ICCV'19	59.4	63.9	63.3	62.2	62.3	57.5	62.8	60.8	58.1	62.5	57.7	59.4	60.9

Coverage of our re-evaluation.

Choe et al. Evaluation for Weakly-Supervised Object Localization: Protocol, Metrics, and Datasets. ArXiv 2020.

Re-ranking of WSOL methods 2016-2019.

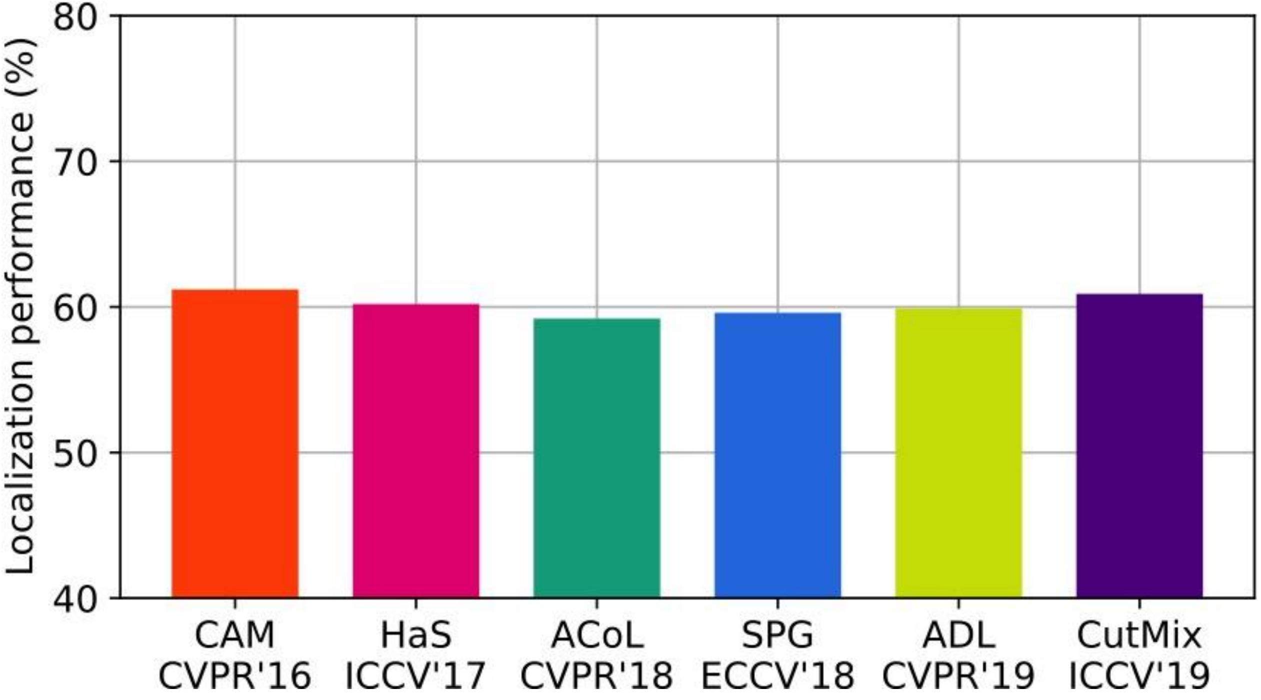


Reported results in respective papers.

Warning: different architectures !!

Choe et al. Evaluation for Weakly-Supervised Object Localization: Protocol, Metrics, and Datasets. ArXiv 2020.

Re-ranking of WSOL methods 2016-2019.

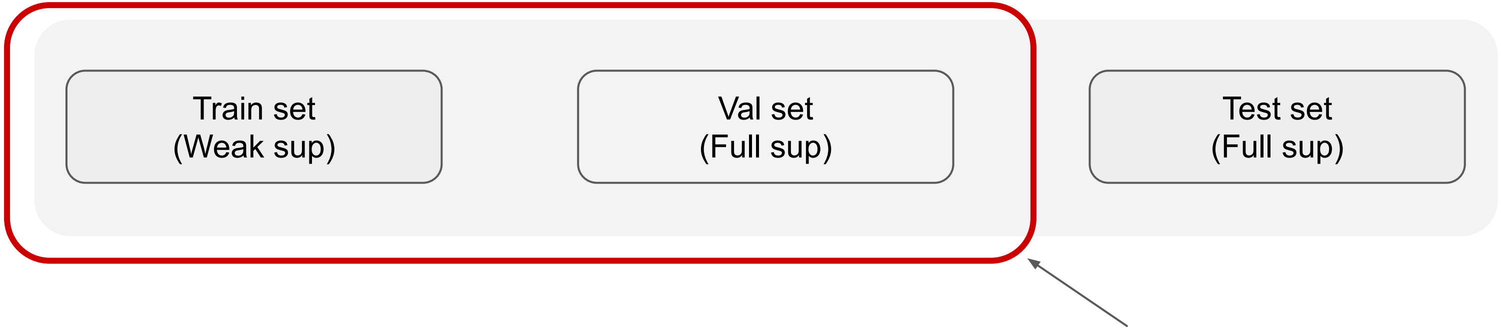


Our re-evaluation.

Mean of ImageNet, CUB, and OpenImages.

Choe et al. Evaluation for Weakly-Supervised Object Localization: Protocol, Metrics, and Datasets. ArXiv 2020.

Using validation set for model training.



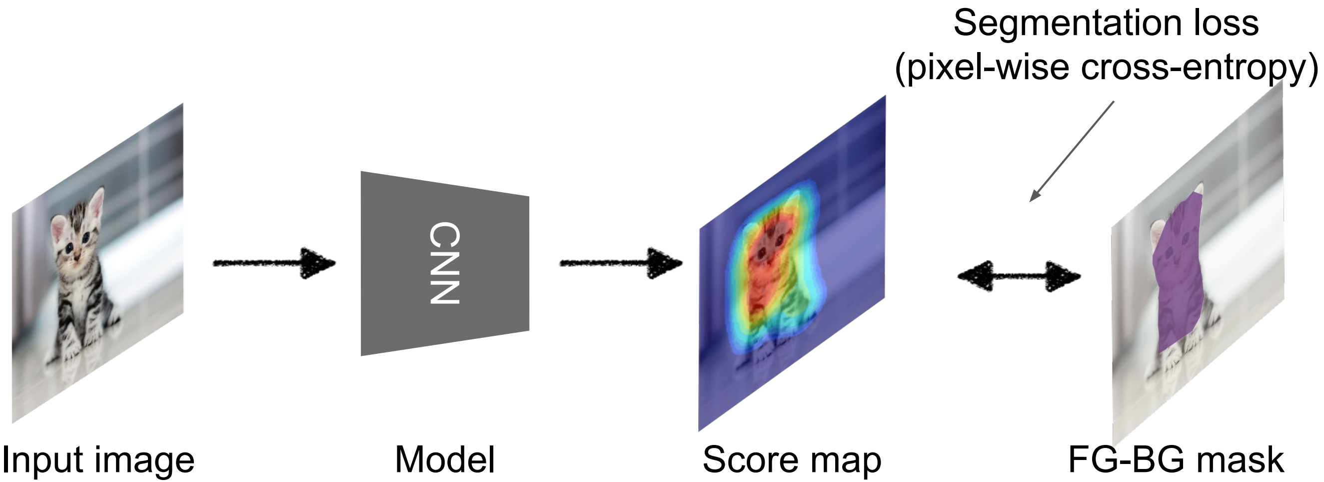
Users are free to use those data which ever way they like.

Val set doesn’t have to be used for validation.

It can be used for model training.

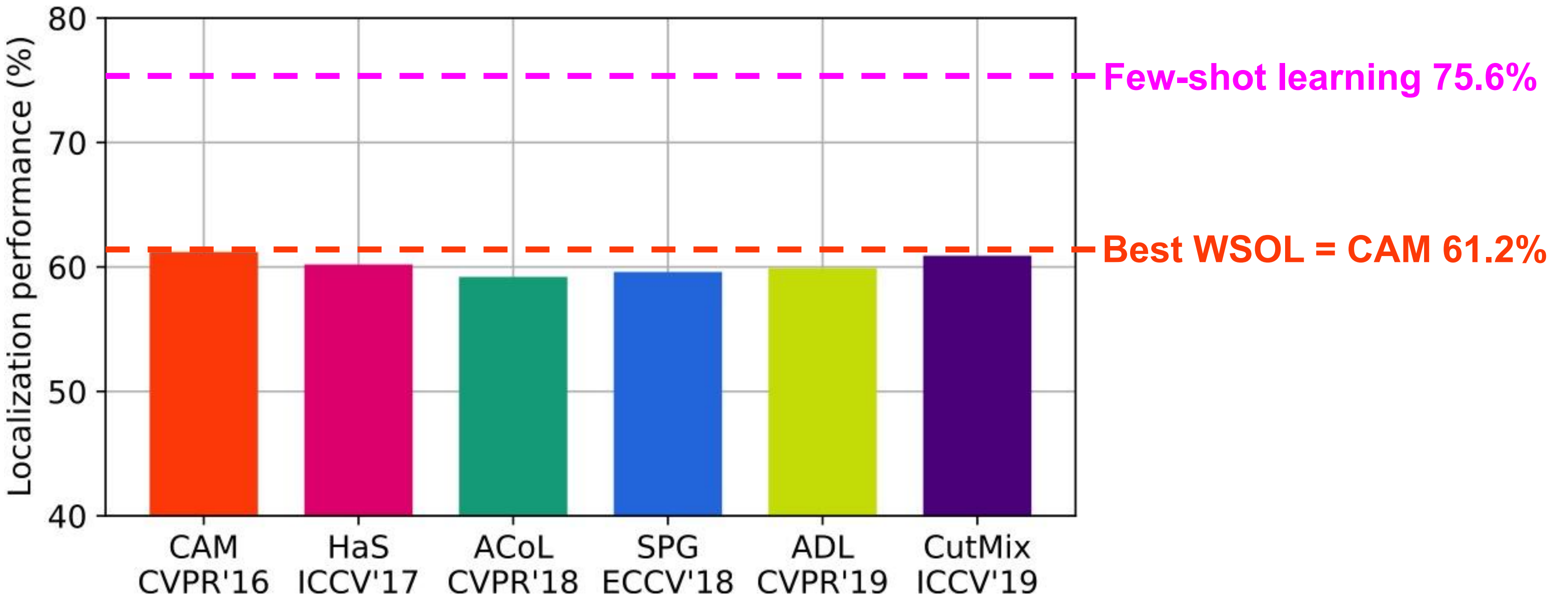
Choe et al. Evaluation for Weakly-Supervised Object Localization: Protocol, Metrics, and Datasets. ArXiv 2020.

Few-shot learning baseline



Choe et al. Evaluation for Weakly-Supervised Object Localization: Protocol, Metrics, and Datasets. ArXiv 2020.

FSL beats WSOL at only 5 samples / class.



Choe et al. Evaluation for Weakly-Supervised Object Localization: Protocol, Metrics, and Datasets. ArXiv 2020.

Implication 1: Complex WSOL methods lost their appeal.

- CAM is simple and effective.
- Few-shot learning is very effective.

Implication 2: New phase of WSOL and WSX research.

Acknowledging the need for extra information opens up new research questions:

- **How to make best use of full supervision?**

  Validation? Model fitting? Or something else?

- **How to exploit existing datasets with diverse supervision types?**

  How to combine multi-modal supervision types?

  OpenImages, COCO, Pascal, ImageNet, Flickr, …

- **Okay we need extra information - but can we minimise it?**

  Maybe under a constraint on the minimal required performance?

Future direction 1: Hybrid weakly-supervised X.

Hybrid-weakly-supervised X Hoffman et al. CVPR’15, Tang et al. CVPR’16

- Combination of different levels and amounts of supervision.

Why relevant?

- Abundance of well-curated and raw data on the web with different levels of supervision. OpenImages, COCO, Pascal, ImageNet, YFCC, Web crawl, …

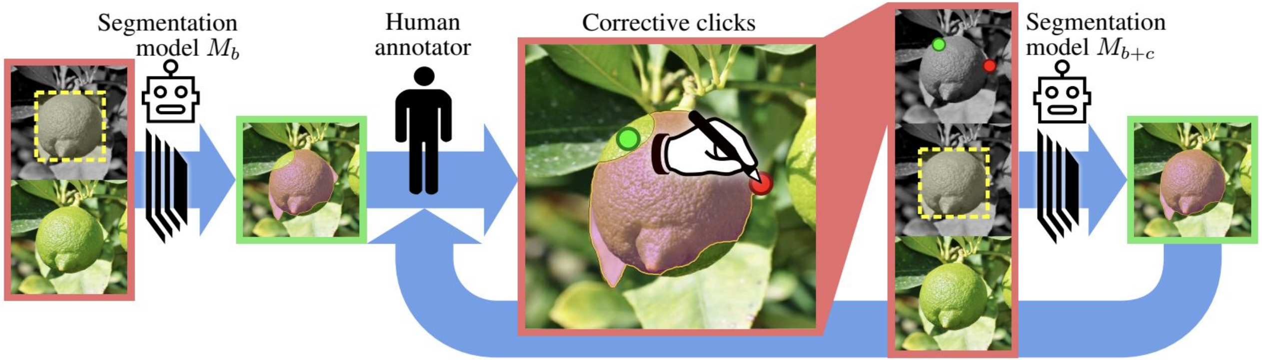
Some non-trivial research questions:

- Setting up the benchmarks.
- Combining multiple supervision modalities.

Future direction 2: Human-in-the-loop.

Another well-defined task is: Minimise the extra annotation, s.t. your method achieves at least *M %* performance.

E.g. Rodrigo's tutorial talk.



Benenson et al. Large-scale interactive object segmentation with human annotators. CVPR'19.

Conclusion and take-aways.

1. WSOL benchmarks are set up like this:

The common strategy for WSOL and other WSX methods is:

(1) introduce **many hyperparameters**.

(2) implicitly tune them with the **full-supervised samples**.

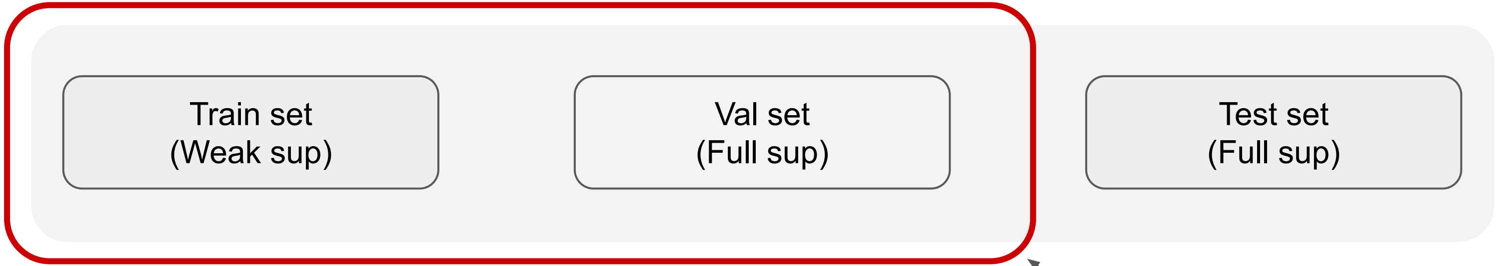
Conclusion and take-aways.

2. This is against the WSOL (and WSX) philosophy, but understandable.

WSOL and many other WSX tasks are ill-posed without extra sources of information or inductive bias.

Conclusion and take-aways.

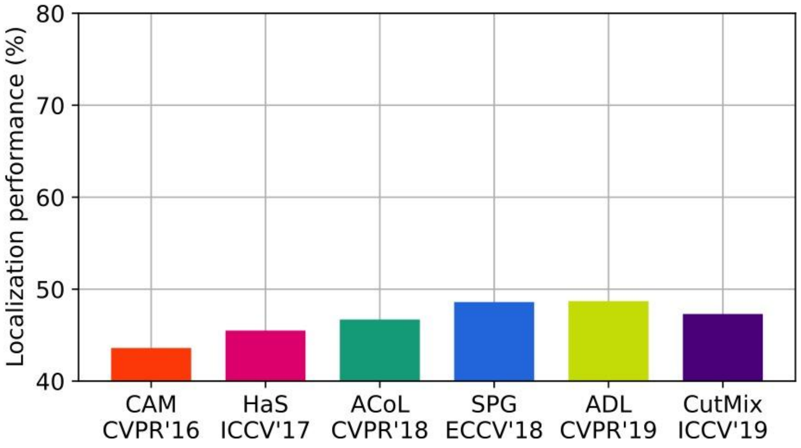
3. Let's legalise the use of full supervision (called "val set").



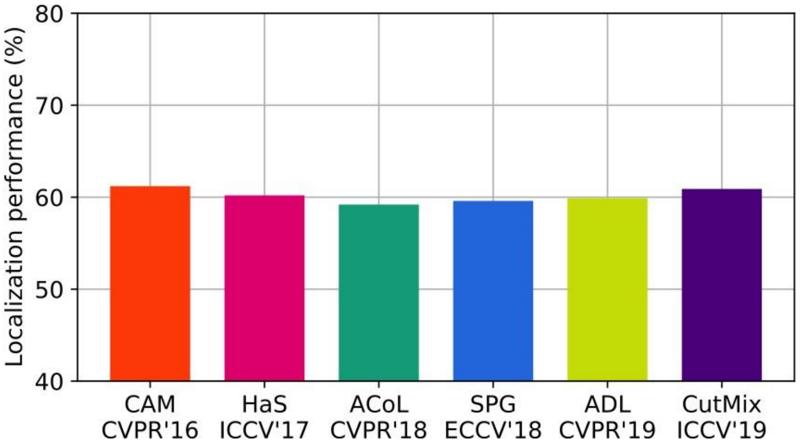
Same amount of full sup ensured for every method.

Conclusion and take-aways.

4. WSX methods can then be compared on the level ground.



Before evaluation clean-up.



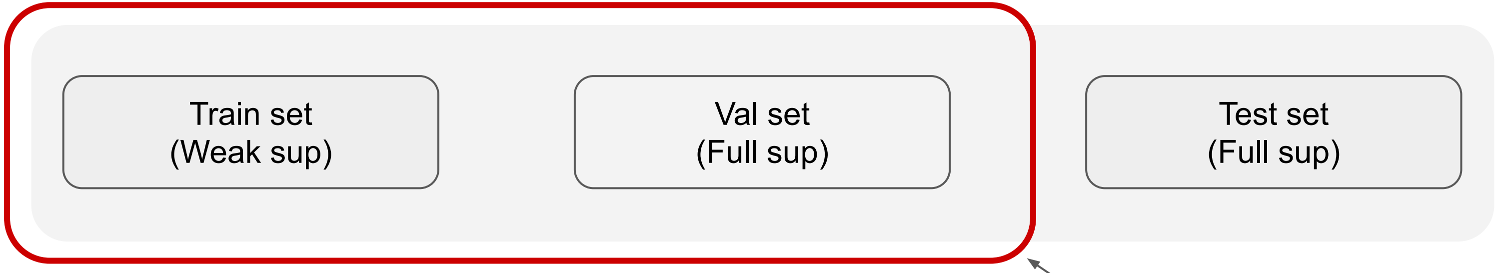
After evaluation clean-up.

Conclusion and take-aways.

5. “Val set” doesn’t need to be used for validation.

This opens up the new phase for WSX research:

- Hybrid weakly-supervised X.
- Human-in-the-loop tasks.



Users can use val set for model fitting as well.